Global Arrays Hands-ON

The DOE Advanced CompuTational Software Collection (ACTS)

Twelfth DOE
ACTS Collection Workshop
Berkeley, California, August 16-19, 2011

Bruce Palmer
Pacific Northwest National Laboratory
Tony Drummond
Lawrence Berkeley National Laboratory

BERKELEY LAB
Lawrence Berkeley National Laboratory

ACTS
COLLECTION

U.S. DEPARTMENT OF ENERGY | Office of Science

Instructions

TUTORIAL Programs

(C and Fortran options available)

Before you start this tutorial:

on carver, type:

```
carver> module load ga/5.0.2
carver> cp $GA_DIR/ga-tutorial11.tar.gz .
carver> zcat ga-tutorial11.tar.gz | tar -xvf -
```

you may want to point your browser to this tutorial:

http://acts.nersc.gov/events/Workshop2011/Talks/GA-hands.pdf

Chose Fortran or C versions of the tutorial problems

- xxxx.c or xxxx.F
- Tutorial programs are incomplete. All you have to do is search file for comments marked with ###, and using the text as hints, replace the comments with subroutines or functions from the GA library to create a working code
- Compile and run as described below
  - Compile your tutorial test program (in **ga-tutorial** directory**)** as follows:
  - Example: If your test program is transp1D-c.tutorial.c, compile the program as follows (*after you make your corrections to the source code*):
    - **`make transp1D-c.tutorial.x`**
- To execute the test, submit using the jobscript in the tutorial directory (remember to source the /usr/common/acts/acts-tuts file once per shell)
  - **`carver> runnow`**
  - **`carver> cd $PBS_O_WORKDIR`**
  - **`carver> mpirun -np 4 ./transp1D-c.tutorial.x`**

Problem 1: 1D Transpose

Transpose a distributed 1D vector containing N elements in the order 1,2,...,N into a distributed vector containing N elements in the order N,N-1,...,2,1

Fortran version of this problem is in the file:

`transp1D.F.tutorial`

C version is in

`transp1D.c.tutorial`.

Working versions of these codes in Fortran and C are in **`transp1D.F`** and **`transp1D.c`**, respectively.

Problem 2: Matrix-Matrix Multiplication

A simple matrix multiply algorithm that initializes two large matrices as GAs. It then multiplies a block of columns by a block or rows from the GAs locally on each processor and copies the result into a third global array.

Fortran version of this problem is in the file
`matrix.F.tutorial`

C version is in
`matrix.c.tutorial`

Working versions are in **`matrix.F`** and **`matrix.c`**

Additional Problems for Hands-On

Both the codes in problems 1 & 2 initialize the data by initializing a local array on processor 0 with all the data and then copying it to a distributed global array. However, for real problems it is usually undesirable to have all the data located on one processor at any point in the calculation.

Can you modify these codes (problem 1 and 2) so that each processor only initializes the data owned by that processor?

From problem1 - 1D- Transpose

Modify code so that each processor only initializes the local array **`a()`** with the data owned by that processor and then copy that data to the global array **`g_a`**

- Hint: Use **`nga_distribution`** and **`nga_put`**

- You will also need to modify the result checking part of the code as well so that it also only uses smaller portions of the total GA

- Hint: copy locally held part of result GA into local array b and corresponding part of original vector into local array **`a`** and compare (use arrays **`lo, hi, lo2, hi2`** to get this data)

From problem2 - Matrix Multiplication

Modify code so that each processor only initializes the local arrays **a** and **b** with the data held locally by that processor. Then copy that data to the global arrays **g_a** and **g_b**.

- Hint: Use **nga_distribution** and **nga_put**

GA Support and References

For GA Help/Support/Bug Report, please send an email to hpctools@pnl.gov

References:
GA Webpage: http://www.emsl.pnl.gov/docs/global/

GA API: http://www.emsl.pnl.gov/docs/global/userinterface.html
(or, go to GA webpage and click User Interface)

GA User Manual: http://www.emsl.pnl.gov/docs/global/documentation.html
(or. go to GA webpage and click documentation)

TO RUN

```
>source /usr/common/acts/acts-tuts
>runnow
>cd $PBS_O_WORKDIR
> type your run commands
> exit
```